Using efficiently autoregressive estimation in Wireless Sensor Networks

Karen Miranda, Victor Ramos, Tahiry Razafindralambo

Using efficiently autoregressive estimation in Wireless Sensor Networks

Karen Miranda[1], Víctor M. Ramos R.[2], and Tahiry Razafindralambo[1]
[1]Inria Lille - Nord Europe, Villeneuve d'Ascq, France
[2]Universidad Autónoma Metropolitana (UAM)
Networking and Telecommunications Research Team, Iztapalapa, Mexico City
{Name.Surname}@inria.fr, vicman@xanum.uam.mx

***Abstract*—Wireless sensor networks (WSNs) are widely deployed nowadays on a large variety of applications. The major goal of a WSN is to collect information about a set of phenomena. Such process is non trivial since batteries' life is limited and thus wireless transmissions as well as computing operations must be minimized. A common task in WSNs is to estimate the sensed data and to spread the estimated samples over the network. Thus, time series estimation mechanisms are vital on this type of processes so as to reduce data transmission. In this paper, we assume a single-hop clustering mechanism in which sensor nodes are grouped into clusters and communicate with a sink through a single hop. We propose a couple of autoregressive mechanisms to predict local sensed samples in order to reduce wireless data communication. We compare our proposal with a model called EEE that has been previously proposed in the literature. We prove the efficiency of our algorithms with real samples publicly available and show that they outperform the EEE mechanism.**

Keywords– Wireless sensor networks; autoregressive processes; data aggregation.

I. Introduction

Applications for wireless sensor networks (WSNs) are of common use nowadays. A WSN consists of a set of sensors to monitor environmental conditions such as temperature, pressure, sound, motion, just to mention a few. The sensor nodes transmit cooperatively their data on the network to a main location. The sensor nodes may be mobile or stationary, they are normally randomly placed on a dynamic environment. WSNs have been used since the decade of 70's in military applications where the main example is the Vietnam War to support enemies detection in areas of difficult access.

The implementation of WSNs poses several challenges due to their low energy resources, limited computing capabilities, and intermittent life, being the former one of the most challenging issues to solve. A common task in WSNs is to estimate the sensed data and to spread estimated samples over the network. Thus, time series estimation mechanisms are vital on this type of processes in order to reduce data transmission and consequently reducing energy consumption. In this paper, we assume a single-hop clustering mechanism in which sensor nodes are grouped into clusters and communicate with a sink by means of a single hop. Thus, we are particularly interested in this type of scenarios rather than on multi-hop scenarios where data must traverse several sensors in order to reach the sink.

The process of gathering information by sensors in a WSN is known as data aggregation. In [1], the authors propose an algorithm called EEE to estimate a time series collected from real traces in order to improve data aggregation. They consider a network arranged in clusters of nodes and focus on temporal aggregation for one hop communication. The aim is to reduce the amount of wireless transmissions by estimating the time series concerned on the WSN with the EEE algorithm. In this paper, we focus on the same one hop communication environment and propose a couple of autoregressive mechanisms to predict local sensed samples in order to reduce wireless data communication. We find that if autoregressive mechanisms are well tuned, important improvements can be achieved on the estimation of real data. We compare our proposal with the EEE algorithm and prove its performance with real samples publicly available. We show that we are able to obtain considerably gains compared to the EEE mechanism.

This paper is structured as follows, in Section II we provide a brief description of time series models and related work on WSNs. In Section III, we describe how we can tune an AR mechanism using the Yule-Walker and the geometric lattice approaches for data aggregation in cluster-based one-hop WSNs. Thus, in Section IV we present the performance measures and the results comparing the performance of the AR algorithms against the EEE mechanism. Finally, Section V concludes the paper.

II. Background and related work

In WSNs the communication task consumes most of the available energy [2]. Hence, one method to reduce the energy consumption is to reduce the amount of messages exchanged between nodes. The goal of data reduction techniques is, precisely, to reduce the data exchanged between the sink and the sensor nodes. Such techniques can be classified as in-network processing techniques, data compression techniques, and data prediction techniques [3].

Data prediction reduces the amount of information sent by building a model of the data evolution. Then, the model predicts the future values with a margin of error. The model is built at the sensors as well as at the sink. If the model is accurate enough, the sink will respond to the users queries without the real data; otherwise, the sensor and the sink need

to retrieve the actual values to update the model. In this paper, we focus mainly on time series forecasting techniques for data prediction.

Before describing the forecasting-based techniques to improve data aggregation, we present a brief introduction to the time series models for a complete understanding of such techniques.

A. Some background on time series models

Time series forecasting methods are commonly used to predict the output values as a function of previous values of a given series. In particular, the autoregressive (AR) model is widely used due to its simplicity and low complexity. This model predicts the value of X_{i+1}, denoted by $\hat{X}_{i+1}$, and is taken as a weighted sum of the last M values of the process X_i. The AR(M) model is defined as follows:

$$\hat{X}_{i+1} = \sum_{m=1}^{M} \phi_m X_{i-m} + \epsilon_i, \tag{1}$$

where ϕ_m represents the coefficients weights, M is called the model's order, and ϵ_i is white noise. There are several approaches to estimate the values of ϕ_m for $m \in [1, \ldots, M]$, such as, Yule-Walker equations, ordinary least squares, maximum entropy estimates, geometric lattice method, and forward-backward method [4].

The autoregressive-moving average model (ARMA) and its generalization on the autoregressive integrated moving average model (ARIMA) are a combination of he AR and the moving average (MA) models. In both, the AR branch represents the dependency between the current value and the M previous values, while the MA branch represents the influence of current and past errors due to white noise on the current and future values.

B. Related work

In [5], Liu et al. propose an ARIMA based approach for data forecasting which consists of two phases. During the preliminary phase, the sink collects enough data to build the ARIMA model from each of its adjacent sensors. At the end of this phase, the sink sends the corresponding coefficient values to the sensors. Then, during the active phase, the nodes predict their values according to the model received from the sink. In order to avoid the model's decay, the sensor sends the k previous real values to the sink when it detects a major change in the data behavior or when the error is beyond a given tolerance. Following the same principle, a forecasting method based on a least mean square filter (LMS) with variable step size (VSS) is presented in [6]. The LMS-VSS method uses the same two phases as presented in [5], also the sink and the sensor use the method locally. The filter is fed with its own estimated values, discarding the real values during the second phase, thereby, reducing the overhead and keeping the consistency of the weights at sink and sensors.

In [7], each node builds its own ARMA model after collecting W samples. Once the model is ready, the sensor sends the model parameters to the sink, then the node collects the next S samples and computes the root mean square error between the predicted and the actual values. If the difference is below the error tolerance, the node will continue to use the current ARMA model, otherwise, the node will build a new ARMA model based on the W recent samples. Thus, the new parameters are sent to the sink. Recently, Said et al. propose in [8] a data aggregation protocol to reduce the energy spent in wireless sensor networks. Moreover, in [9] Bayani and López find that the location of sensors plays an important role in energy consumption in single-sink scenarios.

From now on, we focus only on the algorithm proposed by Ghaddar et al. which we call EEE [1]. Ghaddar in [1] aims to reduce the communication overhead between sensors and their sink by feeding the AR model with its own estimates as samples to generate new estimates. In other words, if $\hat{X}_i$ it is close enough to X_i, the model will use $\hat{X}_i$ as sample; otherwise the sensor will send the actual sample X_i and, based on this sample, the sensor and the sink will recalculate the corresponding coefficients. Furthermore, the authors propose a method to dynamically fit the coefficients weights in Eq. (1), rather than using the traditional methods.

The algorithm proposed by Ghaddar et al. is initialized as follows: at time $t = 0$, $\hat{X}_0 = X_0$, $\phi_m = 1/M$ for all $m \in [0, \ldots, M]$, and the estimation error is given by $e_i = X_i - \hat{X}_i$. Each time that e_i exceeds the error tolerance, the ϕ weights must be adjusted sequentially. Generally, the j-th coefficient is adjusted as follows:

$$\phi'_j = \frac{\left[X_i - \frac{j}{j+1} e_i - \left(\sum_{m=1}^{j-1} \phi_m \hat{X}_{i-m} + \sum_{p=j+1}^{M} \phi'_p \hat{X}_{i-p} \right) \right]}{\hat{X}_{i-j}}, \tag{2}$$

where ϕ' denotes the new value of j-th coefficient $j \in [1, \ldots, M]$. Likewise, the authors use a dynamic error threshold based on previous errors given by

$$\text{thr} = \left(\sum_{i=1}^{M} \left(\frac{|X_i - X_{i-1}|}{M} \right) \right) + \rho, \tag{3}$$

$\rho \in \left[-c\sigma/\sqrt{M}, -c\sigma/\sqrt{M} \right]$ is a random number that represents the uncertainty of the estimation due to data dispersion, σ is the standard deviation of the latest errors, and c is the level of uncertainty, in this case $c = 1.96$ for a confidence interval of 95 %.

III. Autoregressive models comparison

The work shown in [1] proposes an adaptive algorithm to reduce the amount of transmitted data, therefore, saving energy in nodes. This algorithm is based on the AR model adjusting dynamically the model's coefficient with a fixed model's order for all the study cases. The authors use a fixed model's order to simplify their proposal and to avoid taking into account old meaningless values. They claim that the AR(3) model is not efficient since it does not adjust the coefficient values in

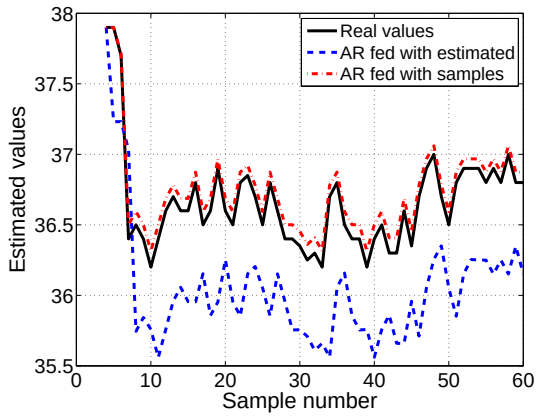

Fig. 1: The error increases when using estimated values as samples compared with the actual values.

terms of relative error. Nevertheless, by analyzing the results reported, there is clearly a bug in the EEE algorithm.

Therefore, we propose to build two AR models based on different methods to fit the coefficients. Considering that we have a set of samples in the past, we calculate the model's order, M as follows. For a given trace, we compute all the values $\hat{X}_i$ for such set of samples, starting with $M = 1$. Then, we increase M by one and repeat the process. The model's order is equal to the lowest value of M preceding an increase in the mean square error. The coefficients ϕ_m in Eq. (1) must be fixed in a way that minimizes the mean square error between $\hat{X}_i$ and X_i by using the Yule-Walker and geometric lattice methods [4].

We consider that by feeding the model with its estimates, we reduce the communication overhead, however, this increases the error between estimated and real values. Figure 1 depicts this increment, here we compare the actual sampled values with the estimated ones, we use the same model with the same order and the same weight but varying the input to estimate the future values. In order to avoid high inconsistencies between the values at the sinks and sensors, it is necessary to add an error threshold. In our approach, each sensor uses Eq. (3) to dynamically calculate a new threshold.

When the threshold limit is exceeded, it is updated and then each node sends the last M errors as well as the last real sample to the sink. Thus, the sink infers the past values from its own estimates, avoiding inconsistencies between the values at the sink and sensors. Besides, the EEE algorithm re-calculates the coefficients with Eq. (2).

IV. Evaluation

In order to validate the accuracy of our estimators for wireless sensor networks, we use the value traces obtained from real time series [10]. We consider that the traces contain the values registered by a single sensor and that should be delivered to their sink. A description of the traces is presented in Table I.

TABLE I: Description of the traces.

Trace	Description	Samples	Min-max values
1	Radioactivity in the ground	1441	100–180
2	Daily morning temperature of adult female	60	38–36
3	Carbon dioxide measurements above Mauna Loa, Hawaii	384	315–360
4	Chemical process temperature readings	226	18–28
5	Heart rate measurements	2568	90–190
6	A garden temperature data	1033	14–28

A. Performance measures

In order evaluate the accuracy of our approach, we focus on the difference between the real data and the estimated data. Particularly, we use the relative error (RE) and the mean square error (MSE). The former reflects the proportional error for each individual value, whereas the latter reflects the overall performance.

Relative error expresses the magnitude of the difference between the real and the estimated values compared to the size of the real value. Recall that X_i is the actual value and $\hat{X}_i$ is its estimation, the relative error is equal to:

$$\mathrm{RE} = \frac{|\hat{X}_i - X_i|}{X_i}.$$

MSE is the arithmetic average of the squared errors and is given by:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{X}_i - X_i\right)^2.$$

We choose to test the accuracy of the algorithms by means of the numeric simulation based on traces. We assume that sink and sensors are in one-hop communication range and both build the AR model given by Eq. (1). At the beginning of the simulation, the value of `thr` is set to 0.05 and we set the model's order to $M = 3$ for the EEE algorithm. For our proposed algorithms, the sensor collects a set of samples first, and then, it computes the order as we described previously, for example for Trace 1 the algorithm chooses $M = 3$ and for Trace 2, $M = 1$. Each model is fed with its own estimated values as samples.

B. Results

We compare three different methods to fit the AR model coefficients, first we use the Yule-Walker method, then the geometric lattice method, and finally the EEE method as in [1] by using Eq. (2). On one hand, coefficient values calculated by mens of Yule-Walker and geometric lattice methods are fixed during the whole session. On the other hand, the coefficients calculated by means of the EEE method change each time that the error exceeds the threshold.

The results for each trace are presented in Figure 2. All the values for relative errors shown Traces 2, 3, and 4 are smaller

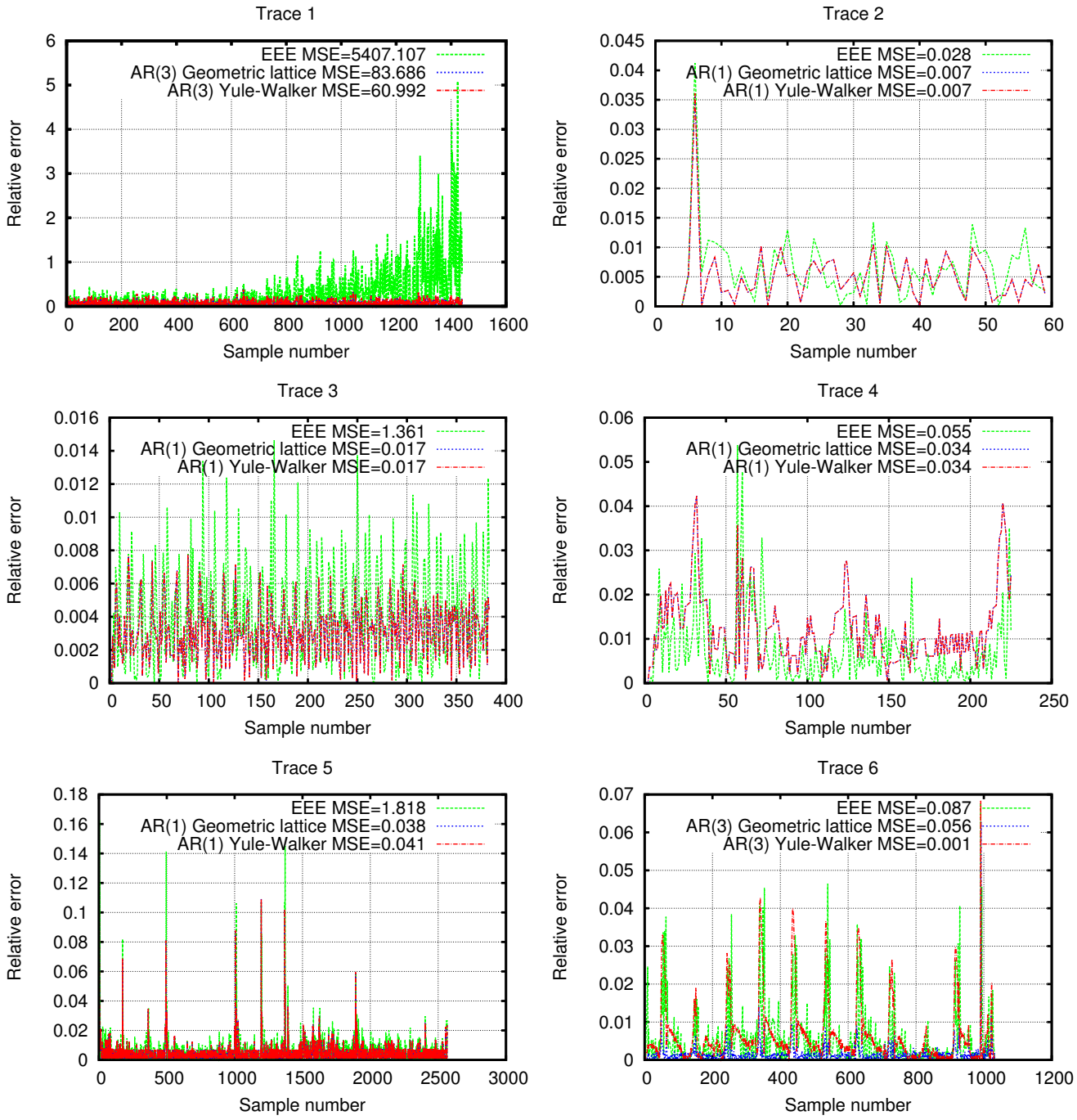

Fig. 2: The relative error of the estimated values for each trace produced by the Yule-Walker, geometric lattice, and EEE methods. The corresponding MSE is indicated as well.

than the original threshold value of 0.05. However, we clearly see that Yule-Walker and geometric methods outperform the EEE method, and we confirm this by regarding the MSE values.

To better understand the behavior in the traces, we also present the linear dependence of samples with themselves and two samples in the past. The sample autocorrelation of Traces 1, 2, and 5 is shown in Figure 3. Regarding Trace 1, we observe a weak dependence between the samples, therefore, for this case an AR(3) model is more suitable. In contrast, for Trace 5 the correlation is strong over several past samples and a small model's order is advisable. To compare the two previous cases, Trace 2 shows a decay in the correlation, and accordingly the AR(1) works well as seen in Figure 2.

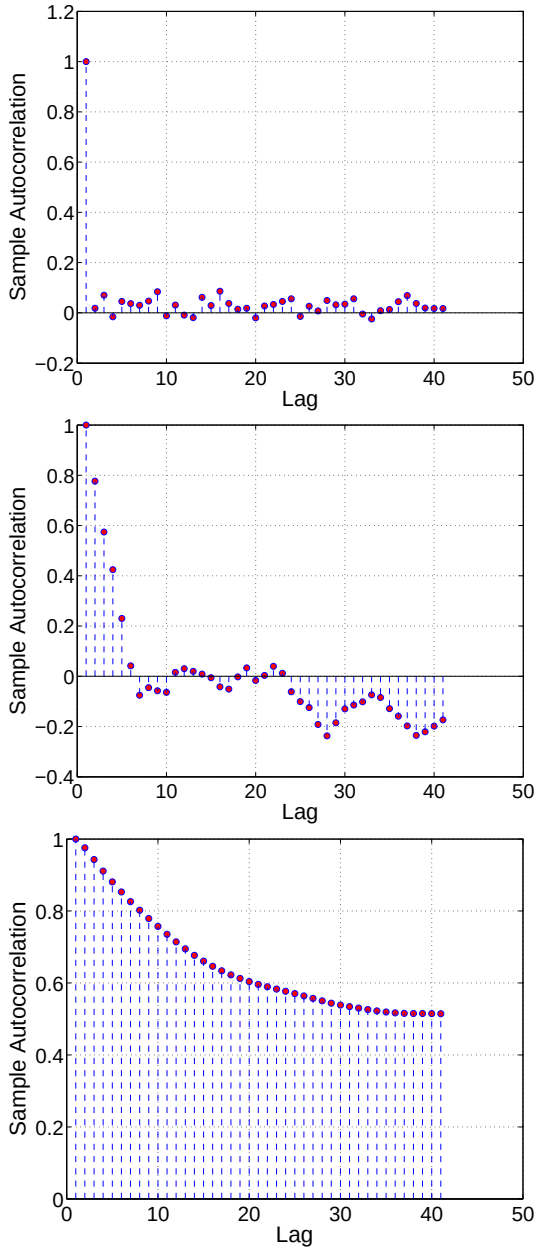

Fig. 3: Correlation function for Traces 1 (top), 2 (middle), and 5 (bottom), the linear dependence of samples with themselves and two samples in the past.

V. Conclusion

We have shown in this paper that a well tuned AR estimator may indeed be used to estimate data series in cluster-based one-hop wireless sensor networks. We showed how an AR process using the Yule-Walker and the lattice-based approaches both exhibit lower relative errors than the EEE model proposed by Ghaddar.

The results we obtained for each of the six traces used showed lower relative errors than their EEE counterparts. We also analyzed how autocorrelation functions are of great help when choosing an AR approach. This is clearly an advantage of the algorithms we propose since the autocorrelation function is included when computing the lags for a given estimation.

We focused on this paper on cluster-based one-hop wireless sensor networks. We believe however that multi-hop WSNs may also be benefited when using the AR processes we tested in this work. Thus, we are interested in studying how this choice may (or may not) impact the performance of such kind of networks.

Acknowledgment

This work was partially funded by the French National Research Agency (ANR) under the VERSO RESCUE (ANR-10-VERS-003) project.

References

[1] A. Ghaddar, T. Razafindralambo, I. Simplot-Ryl, D. Simplot-Ryl, and S. Tawbi, "Towards Energy-Efficient Algorithm-Based Estimation in Wireless Sensor Networks," in *Proceedings of the 6th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, Hangzhou, China, Dec. 2010, pp. 39–46.

[2] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco, "What does model-driven data acquisition really achieve in wireless sensor networks?" in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Lugano, Switzerland, Mar. 2012, pp. 85–94.

[3] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.

[4] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing*. Prentice-Hall, 1996.

[5] C. Liu, K. Wu, and M. Tsao, "Energy efficient information collection with the ARIMA model in wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (Globecom)*, vol. 5, St. Louis, USA, Dec. 2005, pp. 5–10.

[6] B. Stojkoska, D. Solev, and D. Davcev, "Data Prediction in WSN using Variable Step Size LMS Algorithm," in *Proceedings of the 5th International Conference on Sensor Technologies and Applications (SensorComm)*, Nice, France, Aug. 2011.

[7] J.-L. Lu, F. Valois, M. Dohler, and M.-Y. Wu, "Optimized Data Aggregation in WSNs using Adaptive ARMA," in *Proceedings of the 4th International Conference on Sensor Technologies and Applications (SensorComm)*, Venice, Italy, Jul. 2010, pp. 115–120.

[8] A. Mounir Said, A. William Ibrahim, A. Soua, and H. Afifi, "Dynamic aggregation protocol for wireless sensor networks," in *Proceedings of the IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, Barcelona, Spain, Mar. 2013, pp. 356–361.

[9] M. Bayani Abbasy and L. M. López-Ordoñez, "Single-sink mobility performance analysis on a wireless sensor networks," in *Proceedings of the IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, Barcelona, Spain, Mar. 2013, pp. 407–412.

[10] R. J. Hyndman, "Time Series Data Library," accessed on January 2013. [Online]. Available: http://data.is/TSDLdemo